СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

11 - 8555

Д.Д.Арнаудов, Н.И.Янев

ПЕРЕВОД ИНФОРМАЦИИ И ПЕРЕКОМПОНОВКА СТРУКТУРЫ ЗАПИСЕЙ НА МАГНИТНОЙ ЛЕНТЕ ИНИС В ФОРМУ, УДОБНУЮ ДЛЯ ВВОДА В ИНФОРМАЦИОННО-ПОИСКОВУЮ СИСТЕМУ ОИЯИ

1975

11 - 8555

Д.Д.Арнаудов, Н.И.Янев

ПЕРЕВОД ИНФОРМАЦИИ И ПЕРЕКОМПОНОВКА СТРУКТУРЫ ЗАПИСЕЙ НА МАГНИТНОЙ ЛЕНТЕ ИНИС В ФОРМУ, УДОБНУЮ ДЛЯ ВВОДА В ИНФОРМАЦИОННО-ПОИСКОВУЮ СИСТЕМУ ОИЯИ

Арнаудов Д.Д., Янев Н.И. 11 - 8555

Перевод информации и перекомпоновка структуры записей на магнитной ленте ИНИС в форму, удобную для ввода в информационно-поисковую систему ОИЯИ

Описаны алгоритмы перевода информации на МЛ ИНИС в код СДС-6200. Рассмотрена методика организации и перекомпоновки информации в форму, удобную для ввода в ИПС ОИЯИ.

Сообщение Объединенного института ядерных исследований
Дубна 1975

При современном общении между различными организациями как в одной стране, так и в международном масштабе, все больше утверждаются формы обмена информацией, написанной на нестандартных носителях, таких, как перфоленты, магнитные ленты. Типичным примером такого обмена информацией в пределах одной страны является связь между региональными автоматизированными системами управления и обработки информации с подобными системами высшего иерархического уровня. Все чаще создаются и международные информационные системы в определенной области знаний, например, ИНИС /Международная система ядерной информации/. Ежемесячно ИНИС передает участвующим в ней странам и организациям копию всей обрабатываемой ею информации в записи на магнитную ленту.

Актуальным является вопрос ввода, хранения и поиска этой информации в ИПС /Информационно-поисковая система/ ОИЯИ.

Структурная схема ИПС ОИЯИ описана в работе [1]. Формат записи и организация основных информационных массивов сильно отличаются от формата записи и структуры массивов на выходной магнитной ленте ИНИС, что вызывает необходимость перекомпоновки этой структуры. Кроме того, шестибитная символьная сетка СДС /см. *рис. 4*/ отличается от шестибитной символьной сетки ИНИС /*рис. 2*/ тем, что требует предварительного перевода отдельных символов записанной на МЛ информации.

Итак, для того чтобы использовать информацию на магнитной ленте ИНИС как входную информацию в ИПС ОИЯИ, необходимо разрешить две основные проблемы:

1. Перевод символов, описывающих информацию, на МЛ ИНИС.

2. Перекомпоновку и изменение структуры записей и информационного массива на МЛ в форму, удобную для поступления на вход ИПС ОИЯИ.

Для разрешения этих проблем созданы алгоритмы, написанные на языке "ФОРТРАН".

Сначала рассмотрим весь вопрос глобально, т.е. приведем алгоритм, дающий возможность перевода информации, записанной в произвольном коде K_1, в некоторый другой код K_2. Изложенный ниже алгоритм осуществляет перевод произвольного символа, представленного в коде K_1, в соответствующий символ в коде K_2.

Пусть K_2 будет код, который используется в ЭВМ /обозначим ее через E / с размером машинного слова K битов.

Тогда $k=tk_2+\ell$, где t обозначает число символов в одном машинном слове, и $0\leq\ell\leq k_2-1$/для ЭВМ СДС серии 6ООО $k=6O$, $t=1O$, $k_2=6$, $\ell=O$/. Обозначим через ЛЦС /левый циклический сдвиг/ операцию, которая осуществляет следующую трансформацию содержимого одного машинного слова, обозначенного как p_1 / p_2 - число разрядов, на которое производится сдвиг/

для всех $i=1,2,\ldots k$

$i\to i-p_2$, если $i-p_2\geq 1$, а для $i-p_2<1$ $i\to i-p_2+k$, где $i\to i^*/i^*=i-p_2$ или $i-p_2+k$ / означает, что содержимому в i-том разряде машинного слова присваивается i^* разряд. Обычно эта операция осуществляется стандартной программой, включенной в матобеспечение ЭВМ /в СДС-62ОО это программа SHIFT (p_1,p_2), где SHIFT- /сдвиг/ - имя программы, а p_1 и p_2 имеют указанный выше смысл. Если $p_2>0$, то сдвиг будет вправо.

Для осуществления перекодировки необходима еще таблица, находящаяся в памяти E, которая задает соответствие K_1 в K_2/пример такой таблицы приведен ниже/.

Эта таблица /в дальнейшем будем ее обозначать ТАБ/ содержит 2^{k_1} элементов / 2^{k_1} - число символов, которые могут быть представлены в коде K_1 /. j -й элемент таблицы для $j=1,2,\ldots,2^{k_1}$ содержит символ в коде K_2, соответствующий символу в коде K_1, код которого в свою очередь рассмотрен как число в десятичной системе

Таблица

I	2	3	4	I	2	3	4
№ элемента	символ	код	восьмеричная система				
I	пустое	55	00	42	I	II	5I
2	≻	66	0I	43	J	I2	52
3	≠	64	02	44	K	I3	53
4	≡	60	03	45	L	I4	54
5	$	53	04	46	M	I5	55
6	*	63	05	47	N	I6	56
7	≺	67	06	48	O	I7	57
8	↑	70	07	49	P	20	60
9	(	5I	I0	50	Q	2I	6I
I0	)	52	II	5I	R	22	62
II	+	47	I2	52	S	23	63
I2	,	45	I3	53	T	24	64
I3	-	56	I4	54	U	25	65
I4	.	46	I5	55	V	26	66
I5	/	57	I6	56	W	27	67
I6		50	I7	57	X	30	70
I7	0	33	20	58	Y	3I	7I
I8	I	34	2I	59	Z	32	72
I9	2	35	22	60	Λ	6I	73
20	3	36	23	6I	⊥	75	74
2I	4	37	24	62	Γ	62	75
22	5	40	25	63	↑	76	76
23	6	4I	26	64		65	77
24	7	42	27	65			
25	8	43	30				
26	9	44	3I				
27	:	00	32				
28	;	77	33				
29	<	72	34				
30	=	54	35				
3I	>	73	36				
32	→	7I	37				
33	≡	74	40				
34	A	0I	4I				
35	B	02	42				
36	C	03	43				
37	D	04	44				
38	E	05	45				
39	F	06	46				
40	G	07	47				
4I	H	I0	50				

Замечание: В оперативной памяти содержится только 3-я колоика.

счисления, равное $j-1$ /это число используется для входа в таблицу в процессе перекодировки/.

Рассмотрим два конкретных алгоритма.

А1. Этот алгоритм осуществляет перевод K_1 в K_2 /для произвольных k_1 и k_2/, или, точнее, перекодирует содержимое слова А, в котором находятся t' символов в коде K_1, в код K_2. Соответствующие символы в коде K_2 записываются в t' последовательных машинных слов, обозначенных через Т/1/, Т/2/ ... Т/t'/.

Шаг 1. $M = 2^{k-1} + 2^{k-2} + \ldots 2^{k-k_1}$; $i = 1$

Шаг 2. С = АЛМ /Л - знак логического умножения/

Шаг 3. С = ЛЦС /С, k_1 i/, М = ЛЦС /М, $k-k_1$/

Шаг 4. Т/i/ = ТАБ /С +1/

Шаг 5. i=i+1; если $i \leq t'$, то следует переход к шагу 2.

Шаг 6. Выход /М и С - символические названия рабочих ячеек; в шаге 4 С + 1 означает, что к содержимому в ячейке С прибавляется 1/.

Рассмотрим действие этого алгоритма на следующих примерах:

Пример 1.

Пусть в ячейке А в оперативной памяти Е находится слово COMPUTER␣, записанное в коде ИНИС /см. *рис. 2*/. Если Е - это СДС-6200, то $k = 60, t = 10, k_1 = k_2 = 6$, и в восьмеричной системе счисления содержимое А будет:

	43	57	55	60	65	64	45	62	63	00
ИНИС	C	O	M	P	U	T	E	R	S	␣

Шаг 1. В ячейку М засылаем

$77OOOOOOOOOOOOOOOOOO_{(8)}$,

а в ячейку i - 1.

Шаг 2. В ячейке С получится

$$\frac{43\ 0\ \ldots\ldots\ldots 0_{(8)}}{\text{С}} \quad \text{код ИНИС}$$

Шаг 3. Циклический сдвиг в

С 00000 – – – – – $0043_{(8)} = 35_{(10)}$ в М 0077 – – – – – $00_{(8)}$.

Шаг 4. В Т/1/ засылаем содержимое

ТАБ/С+1/=ТАБ/35+1/=ТАБ/36/=

$$= \frac{0 \qquad\qquad 03}{\text{С}} \quad \text{код CDC}$$

Шаг 5. $i = 2 < 10$ переход к шагу 2.

Шаг 2. В С $\dfrac{00\ 57\ 0\ldots\ldots\ldots 0}{\text{Ø}}$ ИНИС

Шаг 3. В С $00\ldots\ldots\ldots 057_{(8)} = 47_{(10)}$

В М $000077\ldots 00_{(8)}$

Шаг 4. В Т/2/ засылаем

ТАБ/47+1/=ТАБ/48/=

$$= \frac{0\ \ldots\ldots\ldots\ldots 17}{\text{Ø}} \quad \text{код CDC}$$

Шаг 5. $i = 3 < 10$ /переход к шагу 2 и т.д./.

Часто бывает желательным результат перекодировки получить опять в исходном слове. При этом использование оперативной памяти минимально, но осуществимо только тогда, когда $k_1 = k_2$.

А2. Этот алгоритм перекодирует и засылает результат в одно машинное слово.

Шаг 1. $M = 2^{k-1} + 2^{k-k_1} + \ldots + 2$, М1 = 0, i = 1.

Шаг 2. С = АЛМ.

Шаг 3. М = ЛЦС /М, $k-k_2$/, С = ЛЦС/ С, k_1 i/.

Шаг 4. С = ТАБ /С +1/.

Шаг 5. С = ЛЦС /С, $k-k_1$ i/.

Шаг 6. М1 = М1VС/ V - знак логической суммы/.

Шаг 7. i=i+1 ; если i<t, то следует переход к шагу 2.

Шаг 8. A=M1.

Шаг 9. Выход.

Пример 2.

На основании примера 1 действие этого алгоритма будет следующее:

Шаг 1. $M=7700\ldots0_{(8)}$, $M1=00\ldots0_{(8)}$, I=1 .

Шаг 2. $C=4300\ldots0_{(8)}$.

C в код ИНИС.

Шаг 3. $M=007700\ldots0_{(8)}$,

$C=00\ldots\ldots043_{(8)}=35_{(10)}$.

Шаг 4. $C=00\ldots\ 003_{(8)}$

C в коде СДС.

Шаг 5. $C=0300\ldots0_{(8)}$

Шаг 6. $M1=0300\ldots0_{(8)}$

Шаг 7. i= 2 переход к шагу 2.

Шаги 2,3,4,5 - действия аналогичны, и т.д.

Шаг 6. $M1=031700\ldots0_{(8)}$

C Ø...

Когда i=11 , имеем в

M1=	O	3	1	7	15	2O	25	24	O5	22	23	55
	C		Ø		M	P	U	T	E	R	S	␣

Шаг 8. Содержимое M1 засылаем в A .

Программаная реализация на ФОРТРАНе

Данные алгоритмы могут быть легко реализованы на алгоритмичном языке ФОРТРАН или PL−1 . Выбор этих языков здесь определен как из-за их широкой популярности среди потребителей ЭВМ, так и благодаря возможности осуществления логических операций.

Ниже представлены программы на языке ФОРТРАН, которые реализуют алгоритмы A1 и A2 при следующем предположении: k= 6O, t= 1O, $k_1=k_2=6$.

Через C1,C2...C64 обозначены однопозиционные символы в коде K_2.

Программа 1.

```
      DIMENSIØN     ITAB (64), IT (10)
1.
2.    DATA (ITAB (I), I=1,64) (1HC1, 1HC2...1HC64)
3.    IM=7700...0B
4.    ID  1 I=1,10
5.    IC=IA, AND. IM
6.    IC=SHIFT (IC, 6 * I)
7.    IM=SHIFT (IM, 54)
8.    IT (I) = ITAB (IC +1)
9. 1  CØNTINUE
```

Программа 2.

```
1.
2. Как и в программе 1.
3.
4.    IM1=0 ... OB
5.    DØ11=1,10
6.    IC=IA. AND.IM
7.    IM=SHIFT (IM, 54)
8.    IC=SHIFT (IC, 6 * I)
9.    IC =ITAB (IC+1)
1O.   IC=SHIFT (IC, 60− 6 * I)
 1    IM = IM1 . ØR. IC
12.   IA=IM
```

Если в матобеспечении ЭВМ отсутствует программа, осуществляющая операции ЛЦС /см. SHIFT /, то указанные выше программы следует немного изменить. Для этого нужно провести операцию деления, т.е. чтобы сдвинуть содержимое слова A на p разрядов вправо /влево/, надо разделить /умножить/ A на 2^p. Например, если $A=100100_{(2)}=36_{(10)}$, то $A/2^2=36_{(10)}/4_{(10)}=9_{(10)}=001001_{(2)}$, т.е. содержание A после деления на 2 сдвинулось на 2 разряда вправо/. В таком случае шаг 3 из $A1$ надо заменить на:

Шаг 3′ $C=C/2^{k-k_1 i}, M=M/2^{k_1}$,

а в алгоритме $A2_1$:

Шаг 3′ $M=M/2^{k_1}$, $C=C/2^{k-k_1 i}$.

Поступающая в ИПС информация /дальше используем термин "Входной массив"/ состоит из переменной длины, первые четыре символа которой содержат длину физической записи */рис. 1/*. Под физической записью подразумевается информация, находящаяся между двумя последовательными промежутками на магнитной ленте.

Информация записана в коде $K1$. Для конкретности будем считать, что это код ИНИС */рис. 2/*. На *рис. 3* показано соответствие между дорожками магнитной ленты и 6-битным кодом ИНИС. Код K_2 будет 64-значным кодом СДС.

Для чтения записи переменной длины вообще можно использовать программу, написанную на языке КОБОЛ. Для этого достаточно указать в выражении DEPENDING ON имя /длину/ того поля записи, в котором находится число символов /в нашем примере это первые 4 символа каждой записи/. Единственная трудность здесь /и, пожалуй, непреодолимая/ состоит в следующем: транслированная программа не осуществляет перевода из одного кода в другой, и поэтому информация, которая содержится в области памяти, обозначенной в нашем примере как ДЛИНА, для СДС-6200 вовсе не будет числом /см. *рис. 2,4/*.

В ФОРТРАНе для осуществления ввода существует два типа оператора READ :

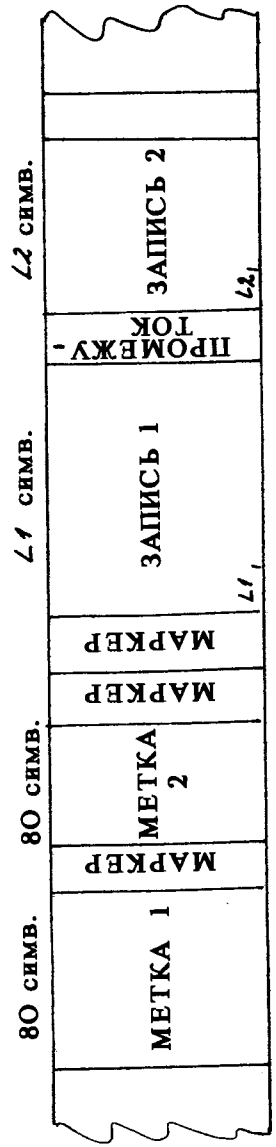

Рис. 1. Организация входного массива на магнитной ленте.

Код ИНИС - 6 битов.

b_6 / b_5 / $b_4b_3b_2b_1$	0 0	0 1	1 0	1 1
0000	L	0		P
0001		1	A	G
0010		2	B	R
0011		3	C	S
0100	$	4	D	T
0101	%	5	E	U
0110		6	F	V
0111	↑	7	G	W
1000	(	8	H	X
1001	)	9	I	Y
1010	*	:	J	Z
1011	+	;	K	[
1100	,	<		
1101	—	=	M	]
1110	.	>	N	↑
1111	/		O	←

Рис. 2.

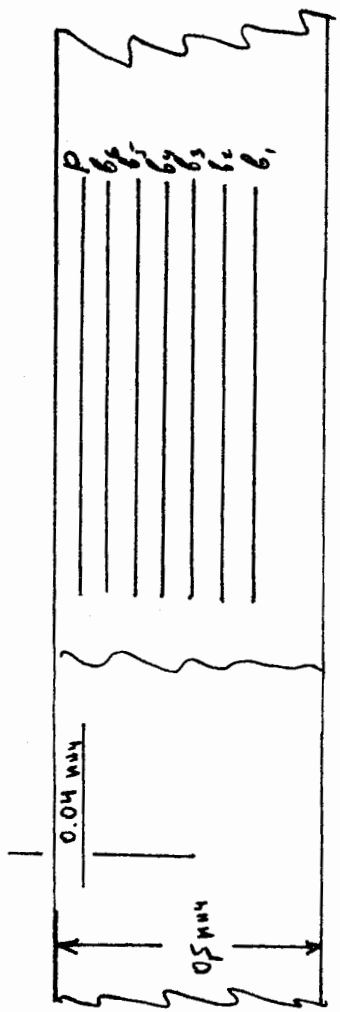

Рис. 3.

P используется для контроля по четности /нечетности/.

а/ READ с форматом,

б/ READ без формата.

Первый из этих операторов нельзя использовать в данном случае по трем причинам:

1. Информация поступает записанной в коде, не совпадающем с тем, который использует конкретная ЭВМ.

2. Число считанных символов зависит от списка переменных, включенных в оператор READ, а это для чтения записи переменной длины непригодно.

3. Тот бит, который на *рис. 3* обозначен через P, используется для контроля по четности или нечетности. Если в поступающей информации используем контроль по четности /нечетности/, а в конкретной ЭВМ - наоборот, тогда все символы /при нормальной работе устройств ввода/ будут объявлены ошибочными.

Последние две причины относятся также и к бесформатному READ .

Итак, для решения задачи ввода в данном примере необходим оператор ввода, который осуществляет чтение одной физической записи, без редактирования вводимой информации и с возможностью переключения по четности или нечетности.

Для СДС-6200, например, это оператор BUFFERIN(i,p), где i - номер логического устройства ввода, p=0 или 1 /контроль по четности или нечетности/.

Для совершения операции ввода необходимо еще указать начало и конец области оперативной памяти, где будет записана вводимая информация.

Итак, полная программа для осуществления ввода информации с магнитной ленты, записанной в коде ИНИС, и ее перекодирование в 64-значный код СДС будет выглядеть так /в программе используется A1 для перекодировки/:

```
1.      SUBRØUTINE  INIS
2.      CØMMØN      IA(205),IT(2048)
3.      DIMENSIØN   ITAB    (64)
4.      DATA (ITAB(I), I=1,64)1H   1HV,1H≠ 1H≡1H$,1H%,
5.      1H  ,1H↑ ,  1H(,1H) , 1H*  ,
6.      .
7.      .
```



```
8.      .
9.      .
10.     .
11.     IM=77000000000000000000B
12.     DØ 2  j=1,205
13.   2 IA(j)= 0
14.     BUFFERIN(1,1)(IA(1), IA(205))
.       IF(UNIT(1).EQ.0)GØ TØ 99
.       DØ  3  j=1,205
.       DØ  4  j=1,10
.       IC=IA(j).AND.IM
.       IC=SHIFT(IC,6*I)
.       IM=SHIFT(IM,54)
.   4   IT(10*(j-1)+I)=IC
.   3   CØNTINUE
.       DØ  5I=1,2048
.   5   IT(I)=ITAB(IT(I)+1)
.
26.  99 RETURN
```

Остальная часть программы включается при достижении конца входного массива.

После использования BUFFERIN для устройства, и прежде чем использовать введенную информацию, состояние операции BUFFERIN должно быть проведено с использованием оператора IFUNIT. Он проверяет, действительно ли ввелись данные в ЭВМ /см. операторы 14 и 26 в приведенной выше программе/.

В рассмотренной программе принято, что максимальная длина записей массива - 2048 знаков. Так как первые четыре символа каждой записи - это длина, то не существует проблем, связанных с интерпретацией информации в записи или, точнее, с определением конца этой информации. Единственное неудобство состоит в том, что каждая

отдельная цифра числа /длина записи/ вследствие перевода расположена в отдельной ячейке памяти. Например: если L = 124O, то после работы подпрограммы ИНИС в ячейках IT(1), IT(2), IT(3), IT(4) оперативной памяти ЭВМ СДС будем иметь /см *рис. 4*/:

IT(1) 3333333333333333334

IT(2) 33.....335

IT(3) 3.....337

IT(4) 3.....33

Чтобы получить длину в 124O символов в ячейке L /типа INTEGER /, то можно использовать операторы ENCØDE| DECØDE с соответствующим форматным списком /с этими операторами можно совершать перемещение информации внутри памяти вместе с редактированием информации аналогично WRITE READ с форматом/. Для указанного примера это можно сделать так:

```
      ENCØDE (10,1,L1)(IT(I),I=1,4)
      DECØDE (10,2,L1)L
1     FØRMAT (10A1)
2     FØRMAT (I4).
```

Вследствие этих операций в L действительно, будет число 124O, и его можно использовать как данное типа INTEGER. .

После окончания алгоритма перевода используется другой алгоритм, написанный на КОБОЛе /он здесь не приложен/, который переформирует структуру записей ИНИС /уже переведенных/ в форму, удобную для ввода в ИПС ОИЯИ. Как результат работы этого алгоритма, создается массив на МЛ СДС с двумя видами записей: первый вид имеет фиксированную длину, второй - переменную.

В этих записях соответствующим образом расположена информация, полученная с МЛ ИНИС и подлежащая вводу в ИПС ОИЯИ.

Алгоритмы реализованы на СДС-62OO и имеют следующую характеристику:

Для перевода и перекомпоновки 4OOO физических

CDC - 64-значный код

знак	код	знак	код
пустое	55	H	I0
≤	74	V	66
%	63	I	II
E	6I	J	I2
→	65	K	I3
≡	60	L	I4
∧	67	M	I5
↑	70	N	I6
↓	7I	O	I7
>	73	P	20
≥	75	Q	2I
¬	76	R	22
.	57	]	62
)	52	S	23
;	77	T	24
+	45	U	25
$	53	V	26
*	47	W	27
-	46	X	30
/	50	Y	3I
'	56	Z	32
(	5I	5	00
=	54	0	33
≠	64	I	34
<	72	2	35
A	0I	3	36
B	02	4	37
C	03	5	40
D	04	6	4I
E	05	7	42
F	06	8	43
G	07	9	44

Рис. 4

записей ИНИС /около 6ООО библиографических записей/ в запись этой информации на МЛ СДС, необходимо 34 *мин.*

Итак, для того чтобы иметь возможность достаточно эффективно формировать основные информационные массивы системы ОИЯИ, созданы алгоритмы преобразования структуры записей и массива с МЛ ИНИС на МЛ СДС-62ОО.

Покажем структуру получаемого массива и формат записей, которые являются входным потоком в систему информационного поиска ОИЯИ.

Для ввода информации используется МЛ СДС, на которой записан последовательный массив, состоящий из логических записей двух типов: первый имеет фиксированную длину, второй - переменную.

Каждая запись заканчивается REC–MARK, т.е. последовательностью пустых символов и закрывающейся скобкой ␣␣␣␣␣␣␣␣␣] (10 символов).

Формат этой записи следующий:

RN	S	T	L	BLANKS	COUNT	TAGINF	RECMARK

RN - это Reference Number, который используется для идентификации данной записи;

S – STATUS CODE

T – TYPE OF RECORD

L – Bibliograficallevel

Blanks - эти разряды не употребляются.

17–20–COUNT дает число знаков записи, следующей за COUNT 1 /, который всегда равен 22О.

21 - 5О – Tag 1
51 - 8О – Tag 3
81 - 11О – Tag 4
111 - 14О – Tag 5
141 - 17О – Tag 6
171 - 2ОО – Tag 7
2О1 - 23О – Tag 8
231 - 24О – Rec–Mark = ␣␣␣␣␣␣␣␣␣]

Все значения полей указаны в работе[3]. Этот тип записей содержит ровно 24О символов. Для каждого поля отведено 3О символов /пока самое большое поле - это "Таг 1", оно занимает не больше 26 символов/. Кроме того, оставлено место на случай, если на выходной МЛ ИНИС появятся и некоторые новые поля /напр. ОО3, ОО5, ОО7/, которые пока не встречаются в выходной информации ИНИС. Запись заканчивается группой символов, образующих поля REC–MARK. Это поле указывает на конец записи.

Во втором типе записей собирается переменная информация, т.е. информация полей с переменной длиной на МЛ ИНИС.

В конце записи также находится REC–MARK. Формат записи следующий:

RN	LEVEL	TAG	BLANK	COUNT	VARTABLEKENGTH	RM

RN – Reference Number ; LEVEL – Bibliographical level; TAG - это номер поля /напр., 1ОО/, чья информацня располагается в VARIABLE LENTH.

BLANK - эти разряды не используются.

COUNT показывает число символов в следующей части записи.

RM – REC – MARK, указывающая на конец записи.

Так как в информационные массивы различные поля записей входят в различных структурах, то это определяет и предполагаемый формат входных записей. Их принадлежность к определенной библиографической записи идентифицируется RN.

Для чтения записей с МЛ СДС используется поле REC–MARK, т.е. читаются все символы до появления REC–MARK. Таким образом, можно работать последовательно с информацией каждого поля, что дает возможность эффективно формировать информационные массивы ИПС ОИЯИ.

Показан фрагмент программы, точнее, описание входного массива на КОБОЛе.

```
FD    FBJBL
      RECORD CONTAINS 21 TO 1550 CHARACTERS
      DEPENDING ON RECORD-MARK
      LABEL RECORD OMITTED
      DATA RECORD IS REC1 REC2
01    REC 1.
      02 RN PIC 9(6).
      02 S PIC 9.
      02 T PIC x.
      . . . . . .
01    REC 2.
      02 RN PIC 9(6)
      02. LEVEL PIC x(3)
      . . . . . .
```

Литература

1. *Д.Д.Арнаудов. Сообщение ОИЯИ, 10-7949, Дубна, 1974.*
2. *IAEA – INIS – 9, Vienna, 1972.*
3. *Д.Д.Арнаудов, В.А.Бирюков. Деп. публ. ОИЯИ, Б4-11-8553, Дубна, 1975.*

Рукопись поступила в издательский отдел
24 января 1975 года.